K-BYTE™, Division of Kay Enterprises Co., is comprised of highly-trained professionals who are dedicated to the development and production of the finest software for the personal computer. Programs of the highest quality are produced for home entertainment, personal education and development, home management and business.

K-BYTE is proud to offer K-DOS™, a superior new Atari* DOS, which is completely compatible with Atari 2.0S and other related software. K-DOS provides you, the programmer, greater reliability, flexibility, and control. K-DOS is command line driven and memory resident with an all important feature of a machine language monitor which allows examination and alteration of memory in hexadecimal and displays ATASCII representation.

K-BYTE and Kay Enterprises reserve the right to make changes in materials and specifications without notice.

*ATARI is a registered trademark of ATARI, INC.

About the Author

Marcus Watts, the innovative author of K-DOS, was born in Princeton, New Jersey in 1961 and grew up in Rochester, New York. He is presently a student at the University of Michigan in Ann Arbor, Michigan majoring in Computer Science. His inquisitive mind has placed emphasis upon how things work, particularly in engineering and chemistry. His interests include music, especially the cello, and reading science fiction.

K-DOS™ USER MANUAL

by

K • BYTE™

a division of

KAY ENTERPRISES CO.

ATD2000D

Printed in U.S.A.

Preface:

This K-DOS Handbook is designed to acquaint and instruct the user with K-DOS, an improved version of a Disk Operating System for the Atari* 800™. The primary purpose is to describe and exemplify the commands necessary to manipulate data to and from the disk drive[s].

SECTION I explains the general contents of K-DOS and lists the system master diskette files. An overall description of K-DOS features is summarized with comparisons and compatibility to Atari's 2.OS.

SECTION II is a handy guide to assist the user with important features, such as terminology, symbols, and syntax conventions used in this handbook.

SECTION III summarizes the procedures for powering up equipment, including the console, disk drive[s], and other peripherals. Memory allocations are suggested for use, and the K-DOS operation is examined.

SECTION IV gives a more detailed listing of the actual features in both the File Management System [FMS] and Disk Utility Program [DUP].

SECTION V details the essential instructions for successfully directing K-DOS commands. These commands are categorized according to type of command for easy usage, i.e. according to disk, file, program, monitor, device, etc. Each category is complete with examples.

The appendices include error messages, FMS patches, a glossary and an index for the user's convenience.

*ATARI is a registered trademark of ATARI, INC.

CONTENTS

SECTION I Introduction

K-DOS, a command-driven DOS, offers more powerful and convenient features than Atari's DOS 2.0S; therefore, it is easier to use. K-DOS permits the user to access disk files and easily manipulate data in numerous ways, i.e. save or load programs, append or delete files, alter memory locations, etc. One of the most significant features of K-DOS includes a machine language monitor which allows the user to examine and alter memory. K-DOS is always memory resident, so it is not necessary to load K-DOS from a disk each time it is used. In addition, K-DOS supports the Atari 850™ handler, the operating system program which allows the use of other devices such as printers and modems.

K-DOS offers the following advantages:

1] Convenience
- [a] K-DOS does not have to be loaded each time it is used.
- [b] K-DOS will persist to load a program until it loads correctly.
- [c] K-DOS defaults filenames and wildcards to give greater adaptability.
- [d] K-DOS allows English commands to be abbreviated.

2] Flexibility
- [a] User Defined Commands allow the user to create personalized commands.
- [b] DOS Character feature permits DOS commands to be executed when the BASIC or ASSEMBLER cartridge is in control.

3] Understandability

Error messages are given in English so they are more readily understood.

4] Control

New powerful commands, such as COLD and LOMem, offer more control over the system.

5] Reliability

Writing large blocks of memory is safer.

The special master diskette accompanying K-DOS includes the following K-DOS system files:

DOS.SYS	FMS & DUP (always memory resident).
TRANS.SYS	TRANS command used to transfer files on a single drive system (UDC).
UDC.SYS	UDC command.
CHERROR.SYS	A file that lists error messages and allows user to change those messages (UDC).
SQUEEZE.SYS	The program SQUEEZE will remove error messages and optionally allow removal of the UDC tables from K-DOS, giving the user additional memory space.
DISKDUP.SYS	DISKDUP command used to duplicate diskettes (UDC).
HELP.SYS	The HELP command (UDC). When running this program (Type HELP or just H), user will get a brief summary of all legal K-DOS commands.
EQUATE.ASM	A system equate file.
DEQU.ASM	An equate file to entry points inside of DOS (global addresses, including user callable subroutines).
HELPFILE.SYS	The file that the HELP command copies.

The master diskette is write-protected for your protection. We recommend that you use DISKDUP (page 13) immediately to make a duplicate of the original, storing it in a secure place where you are not tempted to use it. It is advisable to duplicate (back up) any disk with valuable files to insure against the loss of important information. Write-protecting a disk also prevents you from accidentally writing over and destroying pertinent information. For further instructions on write-protection, see the Atari DOS Manual.

The K-DOS file format is totally compatible with Atari's 2.OS. Optional programs and products you may find useful with K-DOS include:

1] K-COM I, a cartridge-based communications system which turns your Atari into a smart terminal. K-COM I is available through K-BYTE, P.O. Box 456, 1705 Austin, Troy, MI 48099.

2] ASM/ED cartridge by Atari which includes a TEXT EDITOR, as well as an ASSEMBLER and a more sophisticated DEBUGGER.

3] FIX, available through APX, which allows one to recover from certain kinds of disk catastrophies, such as recovering files accidentally erased and "cleaning" a disk whose VTOC (Volume Table of Contents) is erased.

4] Atari Disk Operating System II Reference Manual #CO16347.

5] Atari Disk Personal Computer System Operating System User's Manual #CO16555.

SECTION II Symbols

[ ]	indicates keys on the keyboard	
	[break]	to terminate an operation
	[CTRL1]	to pause output to the screen
	[CTRL3]	to indicate end of file
	[return]	to send input to the computer; press [return] after each command
	[system reset]	to take you back into DOS
	[system reset] [start]	pressed simultaneously with [start] will get you directly into DUP, bypassing the cartridge
{ }	indicates optional parts Ex. WBOOT {n} Proceed {hhhh}	
/	indicates a switch used to modify the action of certain commands	

Command	Switch	Meaning
DISKdup	/All	all sectors
Save	/Append	add data to existing file
DISKdup	/Forever	retry continuously
Run Load	/Map	load map of the records is to be displayed as program is loaded
Run Load	/Noinit	load into memory, but do not initialize
DELete	/Noquery	indicates manipulation of file without asking permission
Run Load	/Patch	ignore memory range error; will then load over DOS
DISKdup	/Put	means that each sector is not checked after it is written
TRansfer	/SIRG	short interrecord gaps
DISKdup	/Write	when destination is written, disk drive checks to insure file was written correctly
,	comma:	optional use in the format of a command
	space:	necessary in commands, particularly when replacing a comma
...	ellipsis:	indicates previous parts may be repeated
*	asterisk:	[1] wildcard: used to replace combination of characters
		[2] locked file: will appear before file in the directory to show it is locked
?		[1] wildcard: used to replace one character at a time
		[2] "this message" see page 14.
———		underlining of commands denotes the proper abbreviation necessary for the successful execution of a command Ex. DELete LOMem
]		denotes minimum abbreviation when defining a UDC
:	colon:	used each time you refer to a device in DOS D: disk drive P: printer
hhhh		denotes a hexadecimal number

SECTION III Powering Up

A. K-DOS boots the same way as Atari's DOS boots.

1. Turn on television set or monitor.
2. Turn on all disk drives.
3. Turn on the Atari 850 (interface module) if you intend to use any peripherals, such as a printer or modem.
4. Properly insert K-DOS master diskette into drive 1 after the BUSY light goes out.
5. Turn on computer console. K-DOS will now boot.

The screen will display the K-DOS version as follows:

K-DOS™ By K-Byte™

Copyright 1981
KAY ENTERPRISES Co.

If you should get a "Boot Error", turn off the computer console for a few seconds, then back on again. Should you continue to get a "Boot Error", check the door(s) of the disk drives and all connections. Further explanation of powering up is explained in the Atari DOS Reference Manual.

B. Memory Allocations

K-DOS requires 14K to load, including 2K for the 850 handler. The utilities (DISKDUP, TRANS) require at least 32K to be useful.

C. K-DOS Executions

K-DOS will:

1. Load itself into memory
2. Initialize itself
3. Boot in the 850's handler
4. Print its title
5. Look for an AUTORUN.SYS file (directs automatic run of a particular file)
6. Enter any cartridge if present or else DOS itself*.

*To enter K-DOS when the cartridge is in control, type "DOS" from the BASIC or ASSEMBLER cartridge, or press (system reset) while holding down (start). K-DOS will respond with a "DOS" and wait for commands.

SECTION IV Features

K-DOS is divided into two parts:

A. FMS - File Management System, the "control" program.

B. DUP - Disk Utility Program, a set of utilities to execute commands called by the FMS. The DUP actually does the labor of the FMS.

A. The FMS in this version of K-DOS offers the following features:

1. The [break] key will stop any I/O with the disk. You no longer need to strike it several times.

2. Writing large blocks of memory is safer because it is no longer written directly from memory. With the Atari FMS, pressing [break] enough times to abort a SAVE from Basic would also destroy the program in memory.

3. FMS will do status checking [check to see if the operation is done correctly or if an error is encountered] of disk drives only on a COLD start. On [system reset], it uses the information it already has.

4. Digits are now allowed for the first character of filenames; the Atari DOS requires the first character of a filename to be alphabetic.

5. Appending a file has been made more efficient. The previous old block is utilized, eliminating wasted space and making the disk usage more efficient.

B. Features of the Disk Utility Program [DUP] are more recognizable than the features of the FMS. They include:

1. Machine language monitor. The Alter/Examine commands take advantage of the screen editor allowing the user to examine and change memory in hexadecimal and display its ATASCII representation. The screen editor may be effectively used because the syntax used for changing locations is the same syntax as printed on the screen.

2. English-like commands with abbreviations. The most common commands may be used with a single letter abbreviation; D,C,B, etc. More dangerous commands, INIT, FORMAT, COLD, DELete, and LOMem require more than one letter abbreviation to decrease the possibility of issuing an incorrect command which could wipe data out unintentionally.

3. New powerful commands: LOMem, COLD. These commands allow more control over the system. LOMem may be used to reserve memory, while COLD, which coldstarts the cartridge, tells BASIC to erase memory.

4. English error messages for errors encountered by DUP and a way to retrieve them. "ERror nn" will display the error message associated with that nn [decimal number]. A list of error messages are given in Appendix A, as printed by the CHERROR.SYS file. Error numbers less than 128 are used for DUP errors. Some error messages are compatible with Atari BASIC.

5. Interception of the BRK instruction, taking you back into DOS. A BRK in Atari DOS will usually crash the system.

6. When the cartridge is in control, [system reset] pressed simultaneously with [start] will get you into DOS, bypassing the cartridge.

7. Easy to use and flexible syntax.
   a. filenames take digits as first character
   b. commas are optional when replaced by a space
   c. lower case input is accepted
   d. commands are abbreviated

8. K-DOS compatability with Atari DOS 2.0S. Users familiar with the Atari DOS can easily adapt to the conveniences of K-DOS.

9. DUP is memory resident. K-DOS does not have to be loaded from a disk each time it is used.

10. UDC (User Defined Commands) permits the user to define a command that loads and runs a machine language program.

11. DC (Defined Character) command. Allows DOS commands to be executed when the cartridge is in control.

12. 850 handler is booted.

13. User callable subroutines. Routines inside of DOS with simple I/O routines to change DOS itself. See DEQU.ASM file.

14. NOTE and POINT work with the screen editor now. POINT is similar to BASIC's Position statement and NOTE is the converse. These are used for consistency in "cleaning" rather than "poking" into memory.

15. The inverse character flag is always reset should you accidentally hit the Atari symbol key.

SECTION V Commands

This section describes in detail K-DOS commands. These commands are grouped into the following categories so that the user may easily refer to and access these commands. Each command is followed by examples to illustrate its function. A summary of commands may be found on page 30.

A. Disk Preperation/Maintenance
Getting disk ready for "storing" data
Maintaining disk for duplication

B. File Control
Manipulation of files

C. Program Control
Management over the systems operations itself

D. Machine Monitor
Direct association with memory

E. Device Control
Management of devices, including peripherals

F. DUP Special
"User Defined Commands"

G. Summary

A. Disk Preparation/Maintenance

Every disk must be formatted before it can be used. "Formatted" means the disk is organized into tracks/sectors so data can be written onto and read from the disk. You may format a disk with existing files, but you must be certain that you do not wish to preserve these files, because they are destroyed when the disk is formatted.

There are 2 commands used to format a disk:

1] INIT n
[n is required for disk number preceded by a space]

2] FORMAT n

1] **INIT n**
Formats a disk on drive n, destroying all previous information, but saves DOS.SYS out.

TYPE: INIT 1
[to format disk on drive 1]

SCREEN DISPLAY: Type Y to format disk 1

TYPE: Y
[press [return] if you do not wish to format disk, otherwise, respond by typing Y]

SCREEN DISPLAY: Saving DOS in D1:DOS.SYS

Ex. TYPE: INIT

SCREEN DISPLAY: Need 1 thru 8 for disk #

2] **FORMAT n**
Formats a disk on drive n, destroying all previous information and reformatting sectors. Recommended for use when you need extra storage, but do not desire the ability to power up with the disk because DOS is not saved. If you desire DOS after you have used the FORMAT command, use WBOOT which is discussed next. Use FORMAT command if you intend to use DISKdup.

TYPE: FORMAT 1
[to format disk on drive 1]

SCREEN DISPLAY: Type Y to format disk 1

TYPE: Y
[press [return] if you do not wish to format disk, otherwise, respond by typing Y]

TYPE: FORMAT 2
[to format disk on drive 2]
FORMAT 3
[to format disk on drive 3]
SCREEN DISPLAY corresponds to the above example for FORMAT 1.

3] **WBOOT {n}**
Command will save DOS as Dn:DOS.SYS on drive n. Use after the FORMAT COMMAND. Disk must be previously formatted before this command can be used. WBOOT may be used on a diskette that already has files without destroying those files.

Ex. TYPE: WBOOT [to save DOS on drive 1]

SCREEN DISPLAY: Saving DOS IN D1:DOS.SYS

Ex. TYPE: WBOOT 2 [to save DOS on drive 2]

SCREEN DISPLAY: Saving DOS in D2:DOS.SYS

4] **DISKdup {scr{,}dest}{/All}{/Write}{/Forever}{/Put}}**

A UDC command used to duplicate the entire contents of a disk onto another disk. This command will run a program, DISKDUP.SYS which copies an entire disk, sector by sector. It does this by using all of free memory as a buffer to read in sectors, and then writes them out. If there is not sufficient memory to hold the entire disk, it will proceed to make more passes.

If only one drive is specified, DISKdup will prompt you to insert the source and destination disks. If arguments are given, they will be fed to the program, or you may type them in at that time. A [return] or [break] in response to "Source,dest " will exit the program.

/All
specifies that all sectors, without regard to the directory, are to be copied. Use this if the disk was not formatted by DOS.

/Write
specifies that when the destination is written, the disk drive is to check that it was written correctly.

/Put
means that each sector is not checked after it is written.

/Forever
means you may retry for as long as you wish. If you should hit [break], or if it should give up a sector, it will stop and respond with "Type C, S, Q, A, or ? for help".

C continue trying
S skip
Q quit after this pass
? this message
A abort immediately

DISKdup will warn you if it could not read or write a sector correctly, and will also tell you how many sectors it copied and how many errors it encountered.

Ex. TYPE: DISK
SCREEN DISPLAY: Disk duplicate V1.8
From [,to]
?

TYPE: 1
SCREEN DISPLAY: Insert source disk, type [return]
?
Insert destination disk, type [return]
?
[The above instructions will be given until disk is copied correctly.]
SCREEN DISPLAY: # sectors copied

Ex. TYPE: DISK 1/F
SCREEN DISPLAY: Insert source disk, type [return]
?
Insert destination disk, type [return]
?
[Instructions repeated]
sectors copied

B. File Control

The following eight commands may be used to manipulate files.

1] Direct
2] Copy
3] DELete
4] LOCk
5] UNlock
6] REName
7] APpend
8] TRansfer

1] Direct {filespec} {,output}

To list the disk directory of the specified files. The filename, extender, and number of sectors will be displayed. The input defaults to D1:*.* and output defaults to the screen editor; E:. Wildcards may be used.

Ex. TYPE: D
Prints a directory of all files on drive 1 to the screen.
D2
Prints a directory of all files on drive 2 to the screen.
D H*
Prints all files whose filenames begin with the letter H. * is used as a wildcard.
D ,P:
Prints a disk directory on the printer.

2] Copy input {,output}
To copy the input file to the output file. The output defaults to the screen editor, E:. To just look at a file, type C filename followed by [return].

TYPE: C TEST.TXT
This lists the file TEST.TXT to the screen. Tokenized Basic programs will appear as mostly garbage.

TYPE: C FILE1 FILE2
To copy FILE1 to FILE2 on the same disk on drive 1.

TYPE: C MONEY D2:EXPENSES
To copy a file named MONEY on drive 1 to a disk on drive 2 and call file EXPENSES.

TYPE: C E:, SWIM
To create small text files. TYPE: C E:, filename [return]. Type in your text.
Breaststroke
Backstroke
Butterfly
Freestyle
[CTRL3]
Press [CTRL3] for end of file. Remember to press [return] before [CTRL3] because the text on the line with the [CTRL3] is not sent to file.

TYPE: C SWIM
To look at a file called SWIM on the screen.

TYPE: C filename
If the file is less than 20 lines, the screen editor can be used as a text editor.

To list the file: Use cursor keys to edit the file. Insert an E: after the C, then enter this line and all others in the file. Press [CTRL3]. File has then been edited and changed accordingly.

TYPE: C PRETTY.ASM,P:
File PRETTY is copied on the printer.

TYPE: C PRETTY.OBJ,N:
File PRETTY is copied to the dummy device. This can be used to verify that the file is okay and can be read.

3] DELete filespec {/Noquery}
To eliminate any file you no longer want on your diskette. You will be asked if you want to delete the file unless the /N switch is specified.

Ex. TYPE: DEL PIC2
To delete file called PIC2
SCREEN DISPLAY: Type Y to delete D1:PIC2 Press [return] to keep file

TYPE: Y [return]

Ex. TYPE: DEL NAME/N
To delete file called NAME without being asked.

SCREEN DISPLAY: DOS

4] LOCk filespec
To lock the indicated file[s]. These files cannot be accidentally deleted or written to until unlocked.

Ex. TYPE: LOC DRIVER.ASM
After file DRIVER.ASM is locked, you will find an * preceding the locked file in the directory. When you attempt to write to a locked file, you will encounter ERROR 167, File Locked.

5] UNlock filespec
To unlock the indicated file[s].

Ex. TYPE: UN DRIVER.ASM
If you want a locked file to become accessible, the UNlock command will reverse the LOCk command so that the file can now be written to or deleted. In the directory, the * no longer precedes the filename.

6] RENname file, filename
To change the name of a file.

Ex. TYPE: REN CHECKS, PAYROLL
To change the name of file CHECKS to PAYROLL on drive 1.

TYPE: REN D2:SUSAN, SUE
To change file SUSAN to SUE on drive 2.

CAUTION: It is not a good idea to give two files the same name.

7] APpend {sourcefile,} destfile
To add data to the end of an old file.

Ex. TYPE: AP DRIVER.ASM,MAZER.ASM
File DRIVER.ASM is added to the end of file MAZER.ASM.

TYPE: AP STATE.TXT
GEORGIA
ALABAMA
TENNESSEE
FLORIDA
[CTRL3]
The source file defaults to E: so the text typed is appended to the destination file when no source file is specified. Small text file is added to STATE.TXT.

8] TRansfer filename {/SIRG} {,filename} {/SIRG}
To duplicate a file on a one-drive system. This command will take a file from the diskette, store it in memory, and then transfer it to another diskette. The program memory is used as a buffer, so it can read the entire file with one read. This is a UDC program in the file TRANS.SYS.

Ex. TYPE: TR PRETTY.ASM
To transfer file PRETTY.ASM from one disk to another, alternating disks several times depending upon the length of the file.

SCREEN DISPLAY: Set up source, [return]
Set up destination, [return]

Ex. TYPE: TR PRETTY,C:/SIRG
/SIRG is used when transferring data to a cassette.

C. Program Control

These commands issue management over the systems operations inclusively; getting back to the cartridge, returning DOS to whatever called it, etc.

1] Back
2] WARM
3] COLD
4] Xit
5] UNLOAD
6] LOMem
7] DC {character}

1] Back
This is the official way to get back to the cartridge, BASIC or ASSEMBLER cartridge. If BASIC, then BASIC is in control.

Ex. SCREEN DISPLAY: DOS

TYPE: B

SCREEN DISPLAY: READY
If you have the BASIC cartridge inserted.

or

SCREEN DISPLAY: EDIT
Takes you back to the ASSEMBLER cartridge.

or

SCREEN DISPLAY: No cartridge
When cartridge has not been inserted.

REMEMBER: To get back to DOS, type DOS or press [system reset] and [start] simultaneously.

2] WARM

To force a warm start, to reinitialize without changing memory, to close files, to reset pointers without erasing memory. Use only if you think DOS might be confused about the cartridge. [This command is useful after RESET command, when you are certain cartridge's memory is intact.]

Ex.

SCREEN DISPLAY:	DOS	DOS
TYPE:	WARM	WARM
SCREEN	READY [BASIC]	READY [ASSEMBLER]

3] COLD

To coldstart the cartridge. Like NEW in BASIC or in the EDIT/ASM. but more thorough because it erases the program area [user area] of memory.

Ex.

SCREEN DISPLAY: DOS

TYPE: COLD

SCREEN DISPLAY: Type Y if okay to coldstart cartridge?

TYPE: Y

SCREEN DISPLAY: EDIT [ASSEMBLER]
READY [BASIC]

4] Xit

Tells DOS to return to wherever it was executing: Another way to get back to the cartridge. In BASIC, if DOS was called from a program, the program will continue.

Ex.

SCREEN DISPLAY: DOS

TYPE: X

SCREEN DISPLAY: READY [BASIC]

5] UNLOAD

Tries to erase area where cartridge is; unloads any RAM based cartridge and resets LOMem back to the end of DOS. Program inserted between DOS and LOMem area is erased.

Ex.

SCREEN DISPLAY: DOS

TYPE: UNLOAD

SCREEN DISPLAY: Type Y if ok to coldstart cartridge?

TYPE: Y

SCREEN DISPLAY: DOS

6] LOMem {hhhh}

Sets the bottom of memory for a cartridge. This can be used to reserve memory for a machine language subroutine that you do not want the cartridge to "play" with. Since this means that the cartridge's memory space has been moved, the cartridge is cold started. If {hhhh} is omitted, a summary of memory usage is printed.

Ex. TYPE: LOM

SCREEN DISPLAY:	DOS	Bottom Low	High	Top
	2F58	31D8	31D8	31D8	9C1F

Ex. TYPE: LOM 2F58
Low memory must be at least 31D8. 2F58-31D7 is used for disk buffers.

SCREEN DISPLAY: LOMem out of range.

7] DC {charactar}
Allows user to define a character, such as a "/", and when character is defined, the DOS commands may be used with the cartridge.

Ex. TYPE: DC /
SCREEN DISPLAY: DOS

TYPE: B
SCREEN DISPLAY: READY

TYPE: /C PRETTY
The default is ",". DC with no character turns the feature off. Use WBOOT to save this character on the disk if you always want your character to be different than a ",". The DC character by itself puts you somewhere between BASIC and DOS. [CTRL3] takes you back to BASIC. Type "DOS" to get back to DOS.

D. Machine Monitor

The following commands allow the user to deal directly with memory; to examine memory, to change memory, etc.

1] Run
2] Load
3] Save
4] Go
5] Proceed
6] Examine
7] Alter
8] REgister

1] Run file {/Map}{/Noinit}{/Patch}
To load an object file and run it. If the program loads over the program area, the loader will ask you if you want to coldstart the cartridge.

/Map
denotes a load map of records is displayed as it is loaded.

Ex. TYPE: R PRETTY.OBJ/M

/Noinit
specifies that file may be loaded in memory, but do not initialize. This switch will prevent a normal LOAD to run this program.

Ex. TYPE: R PRETTY.OBJ/N

/Patch
specifies that memory range error is to be ignored. Pointers will load in where file instructs it to be loaded. It will then load over DOS.

Ex. TYPE: R D2:HERE/M/N
SCREEN DISPLAY: 6000-6090
02E0-02E3
6010 INIT
6020 GO
BRK at 6020

2] Load file {/Map}{/Noinit}{/Patch}
To load a file into memory. It can be run with the Go command, if it has a run address (at $2E0).

/Map
denotes a load map of records is displayed as it is loaded.

/Noinit
specifies that a file is NOT to be initialized.

/Patch
ignore memory range error; load in where file instructs.

Ex. TYPE: L D2:HERE/M/N
SCREEN DISPLAY: 6000-6090
02E2-02E3
6010 INIT
6020 GO

3] Save file {/Append} beg end {{init} start}

To save memory on a disk file. Locations $2E2 and $2E3 will be set to the run address for the Run command. All addresses are hexadecimal.

/Append
Adds data to the object file without writing another object file header.

Ex. TYPE: S D2:HERE 6000 6090 6010 6020

4] Go {hhhh}

To start execution at the indicated hexadecimal address (or at the last loaded or saved file's run address). Note that this command does an implicit CLOSE command and doesn't load the registers with their stored values. A return address is left on the stack so RTS will return control to DOS. This may be used to restart UDC.

Ex. TYPE: G 5000

5] Proceed {hhhh}

To continue execution from a BRK instruction. Change the PC if {hhhh} is specified. This command does not change the registers, and does NOT close files. It can be used with the BRK instruction and the Alter command to set breakpoints to debug a machine language program.

Ex. SCREEN DISPLAY: Brk at 0602
TYPE: P or P 403E

3] Examine {◄first►{,◄last►}}

To look at memory in hexadecimal and ATASCII. The format is: addr◄h1 h2 h3 h4 h5 h6 h7 h8 "12345678, compatible with the Alter command. The Examine command defaults to the last of the following: after last Examine command, last Alter command, last loaded program or run address if present. Examine n will report 8 bytes starting at n.

Ex. TYPE: E 5000,5010 or
E 700 or
E

7] Alter { hhhh}◄{hh}{,}{hh}{,}..... garbage
Alter {hhhh} ◄"ascii
or [implicit mode] {hhhh}
[same as above]

NOTE: To change memory in hexadecimal or ATASCII. $60, a diamond or grave accent (non-displayable characters) on the printer cannot be deposited in memory because the Examine command uses this to indicate a byte that is not a displayable ATASCII code.

Ex. TYPE: A 600◄0 or 600◄0 or 600◄0 "$#?3T

8] REgister
REgister {r◄h}

To examine and alter the saved 6502 registers. RE examines all the registers. RE r◄h alters registers. r represents A,X,Y,S,C,P as follows:

A,X,Y	are the corresponding registers
S	is the stack pointer
C	is the flags register
P	is the program counter
h	is a hexadecimal quantity (1 byte, except for P which is 2 bytes)

NOTE: This command is intended for those with the knowledge of machine language.

Ex. TYPE: RE P◄9AED
RE A◄9B, X◄ED or
RE

E. Device Control

These commands regulate the functions of the devices, such as the screen editor, the disk drive[s], a printer and/or interface module.

1] RESET
2] Text
3] CLose
4] ERror nn

1] RESET *

This command resets all devices that DOS recognizes. It also coldstarts the cartridge. It can be used while setting the disk drives, and the number of file buffers. To do this, type:
Alter 709◄#buffers, drives
RESET
[drives is a bit map of the drives that you want and # buffers is the maximum number of I/O channels that you intend to have open at the same time to the disk]. This does not kill any user devices. It is most useful when changing buffers.

***CAUTION:** Do not confuse the RESET command with the [system reset] key.

Ex. TYPE: RESET
SCREEN DISPLAY: Type Y if ok to coldstart cartridge?

TYPE: Y
SCREEN DISPLAY: DOS

2] Text

This command rewrites the display list and causes the computer to display a clear text screen. It reopens the screen editor in mode 0 and is equivalent to GR.0 from BASIC.

Ex. TYPE: T
SCREEN DISPLAY: DOS

3] CLose

To close all open files, turn off the sound, reset the vertical blank vectors, and turn off the player missile graphics. It is similar, yet more powerful than the BASIC command END. BASIC will automatically close files before it calls DOS.

4] ERror nn

This command displays the error message corresponding with nn, a decimal number. Numbers less than 128 are used by K-DOS errors.

Ex. TYPE: ER 144
SCREEN DISPLAY: ERROR 144, DEVICE ERROR

Ex. TYPE: ER 38
SCREEN DISPLAY: Incompatible disk drive

NOTE: A list of error messages may be found in Appendix A.

F. DUP Special Commands

These four commands offer special privileges for the Disk Utility Program.

1] UDC
2] Ident
3] KILL
4] REVIVE

1] UDC User Defined Command

A UDC is a command that permits the user to define a command that loads and runs a machine language program.

The UDCs supplied on the system master diskette along with DOS include:

TR]ANS,	D:TRANS.SYS
H]ELP,	D:HELP.SYS
U]DC,	D:UDC.SYS
CH]ERROR,	D:CHERROR.SYS
DISK]duplicate,	D:DISKDUP.SYS

] denotes minimum abbreviation when defining a UDC. When deleting a UDC, you may use the abbreviation, but not the].

A UDC can exit with a BRK instruction or an RTS if the stack is preserved. You should use WBOOT or INIT to save the copy of DOS with the UDC table to the disk. DOS commands take precedence over UDCs.

Ex. TYPE: UDC
SCREEN DISPLAY: UDC manager V1.2
List, Add, DELete, INIT, Stop
?

L	List the UDC table
INIT	Clear the UDC table
DEL cmd	Delete the command from the UDC table
A cmd {,} file	Add the command to the UDC table
Stop	Halts the UDC program

2] Ident

A command to identify the version of DUP that is in use, repeating the K-DOS title.

Ex. TYPE: I
SCREEN DISPLAY: K-DOS™ By K-Byte™ [same message as when DOS is booted]
Copyright 1981
KAY ENTERPRISES Co.

3] KILL

KILL deletes the DOS E: vectors and serial input/output patch. Useful if your program machine language accidentally wipes DUP out.

DOS intercepts screen editor and serial input/output. If DUP program has been interfered with and will not run properly, KILL will prevent the use of the program.

Ex. SCREEN DISPLAY: DOS

TYPE: E 20A
SCREEN DISPLAY: 020A◄11 E8 E3 27 D1 EA B2 E7
DOS

TYPE: KILL
SCREEN DISPLAY: DOS

TYPE: E 20A
SCREEN DISPLAY: 020A◄11 E8 90 EA D1 EA B2 E7

4] REVIVE

REVIVE is the opposite of KILL. All errors used in DUP are equated in EQUATE.ASM. REVIVE allows DOS to accept, rather than intercept, the screen editor and serial input/output.

COMMAND SUMMARY

Disk Maintenance	INIT n
	FORMAT n
	WBOOT {n}
	*DISKdup {scr{,}dest}{/A}{/W} {/F}{/P}}
File Control	Direct {filespec}{,output}
	Copy input {,output}
	DELete filespec {/N}
	LOCk filespec
	UNlock filespec
	RENname file,filename
	APpend {sourcefile,} destfile
	*TRansfer filename {/SIRG} {,filename}{/SIRG}
Program Control	Back
	WARM
	COLD
	Xit
	UNLOAD
	LOMem
	DC {character}
Machine Monitor	Run file {/M}{/N}{/P}
	Load file {/M}{/N}{/P}
	Save file {/A} beg end {{init} start}
	Go {hhhh}
	Proceed {hhhh}
	Examine {◄first►{,◄last►}}
	Alter {adr}{◄} hex....or "ascii
	REgister {r◄h}
Device Control	RESET
	Text
	CLose
	ERror nn
DUP Special	*UDC
	Ident
	KILL
	REVIVE

_____ Indicates the minimum abbreviation.

* Indicates a UDC command that normally resides in a disk file.

APPENDIX A Error Messages

Error No.	Error Name	Cause and Recovery
1	Illegal command	Type HELP for help. DOS didn't understand that command.
2	Not enough memory	The machine language program loads at too high a location, or the UDC table overflowed, or TRANS didn't have enough memory to load the whole file.
8	Number?	You typed an illegal number. Periods and + or – signs are never legal in DOS. A-F is allowed for hexadecimal numbers only.
12	Go where?	You didn't specify an address to go to, and there is no run address from the last file loaded. Note: [system reset] destroys the run address.
21	Bad load file	You tried to load a file that isn't a legal load file. Try specifying an extension. Note "File." will specify a null extension.
32	Syntax?	There are extra or illegal parameters.
33	Switch?	Used incorrect Switch designator.
35	Filename too long	You typed in an illegal filename. See "filename" in Glossary.
36	Not a disk file	You can only delete files on a disk drive.
37	No cartridge	B and X commands will work only when you have inserted a cartridge to which you may return.
38	Incompatible disk drives	You can not back up an 810 disk on an 815.
39	Need 1 thru 8 for disk #	Disk drives are numbered from 1 to 8.

40	Illegal User Def'd Command	You tried to delete a UDC command that was not in the table.
41	Not Basic—use Back	DOS cannot load or run Basic programs. DOS only knows the internal format of machine language files, and those saved by DOS or the ASM/ED CARTRIDGE.
42	LOMEM out of range	You specified an illegal address for the LOMem command.
43	Can't overlay DOS	You tried to load a file that loaded where DOS is.
44	Can't proceed	
128	**Break	You hit [break] key. Will stop execution.
130	No such device	You have tried to use an undefined device. Check for the correct device.
136	End of file	No more data is listed in your file.
138	Device timeout	You have issued an incorrect device number or specified the wrong device. Examine all connections. Check and retry the command.
139	Device NAK	No response because of bad parameters. Device may have received bad data from the computer.
141	Cursor out of range	Cursor is out of the range for the mode you selected.
144	Device error	This device cannot execute a legal command. Check if disk is write-protected.
146	Funct. not implemented	The function is not contained in the handler. You are trying to use incompatible commands and devices.
154	Concurrent mode I/O not active	See 850 Handler Manual.
160	Bad drive num	Drive numbers must be 1 thru 8.
162	Disk full	There are no more free sectors on your diskette. It is time for another diskette.
164	File overwritten	Sector does not contain information from this file.
165	Bad file name	The filespec you have used has incorrect characters in it. See Glossary for correct file-specification.
167	File locked	You cannot append or delete a locked file.
169	Directory full	All the space in the directory has been used.
170	File not found	File does not exist.
172	Incompatible DOS format	File not created by DOS 2.0S or K-DOS.
173	Can't format disk	Bad sectors have been encountered, so disk cannot be formatted.

APPENDIX B FMS Patches

The following list of patches may be used to change the FMS allowing you to recover files, etc.

NOTE: These changes are reserved for the advanced programmer. Use with caution!

"ALTER 41◄0" from 3. This tells SIO to be quiet, so any I/O over the serial bus will be silent, including using the disk drive or printer. This is reset by [system reset] to 3. "POKE 65,0" can be used in Basic programs.

"ALTER 792◄0" from 3. Use this to change the retry count inside of FMS from 3 times to 256 times. This is helpful if the disk is hard to read.

"ALTER 77C◄3" from F. Disk normally times out after 15 seconds. This changes that to 3 seconds.

APPENDIX C Glossary of Terms

Addr — Abbreviation for address of memory location.

Arguments — Variables listed in filename and in {} after the filenames; everything after the command.

ASCII — The American Standard Code for Information Interchange.

Byte — 8 bits; basic unit of measurement.

Boot — A subroutine which initializes the program as computer is powered up.

Buffer — Temporary holding area for data which may be further processed. K-DOS has an internal 256 byte buffer for certain commands [Copy, DELete, Direct].

CC — 6502 program status byte; the Flags Register.

CIO — Central input/output subsystem.

D: — Device reference to disk drive.

Defaults — Conditions of falling through if output is not specified; K-DOS has a series of defaults so that you don't have to specify common parts:

nothing	→	D:
n	→	Dn:
filename	→	D:filename
:filename	→	D:filename
d:filename	→	d:filename
dn:filename	→	dn:filename

Where n is a single digit, d is a single letter for a device name, and filename consists of a name of up to 8 alphanumeric characters, and an extension of up to 3 characters.

Certain commands, Direct, Load, Run, DELete [but not with /NOQUERY] have wildcard defaults. This is done after the previous defaults are done, and consists of:

D:	→	D:*.*	all files
D:	→	D:name.*	all namefiles with extensions
D:name.	→	D:name.	extensions only
D:name.ext	→	D:name.ext	just that file

Dest Abbreviation of destination, i.e. destination file, the receiving file during a transfer of information.

DUP Disk Utility Program.

E: Device reference to the screen editor.

Filename Alphanumeric characters assigned to identify a particular file; up to 8 characters plus 3 additional characters in the extension.

Filespec File specification consisting of 1 character device name, an optional device number, a colon, a filename up to 8 characters and optional extension [consisting of a period followed by up to 3 characters].

Flags register 6502 status register

FMS File Management System.

K Kilobyte; 1024 bytes of memory.

Lower case Indicates parameters for the commands; K-DOS accepts lower case input.

N: Dummy device in K-DOS; anything written to it disappears without a trace; sends return to end of file.

n Represents single digit.

nn Represents decimal number, i.e. ERror nn.

NOTE Retrieve I/O device's place.

Object Code Another name for machine language.

Object File A file with object code in it; DOS can load and generate files that work with DOS 2.0S and the ASSEMBLER cartridge.

Parameters Variables which give additional information about a command, i.e. filenames, disk drive numbers, hexadecimal numbers.

Patches Repairs used to fix mistakes; see Appendix B.

PC 6502 Program counter which indicates the location in memory where computer was executing program.

POINT Set I/O device's place.

POKE To alter a memory location in BASIC.

PEEK To examine a memory location in BASIC.

ROM Read Only Memory; permanent memory storage which cannot be changed.

RTS ASSEMBLER instruction; return from subroutine.

Scr Abbreviation of source, as in source file; the file containing the information to be sent to the destination file.

SIO Serial input/output.

SIRG Short interrecord gaps referring to cassette tapes; see TRansfer command.

Stack pointer 6502 stack pointer; indicates current entry point of a stack of information.

Syntax The rules of commas, characters, notations, etc. necessary to properly execute a command.

Text Files Units of information, i.e. lists, results, copies, which may or may not be a program.

Tokenizing Process of converting BASIC instructions into symbols; for example, "Run" is reduced to 1 symbol or byte.

UDC User Defined Commands permit the user to define commands that run machine language programs.

Upper Case Indicates parts necessary for input. For example, in the command Copy, the C is the only character needed for proper execution. With DELete, DEL is required.

Vectors Pointers in memory, often to interrupt code.

VTOC — Volume Table of Contents; bit map of all available sectors.

Wildcards — "?" and "*" – wildcard characters.
"?" will replace any single character.
"*" will replace multiple characters (rest of filename).

INDEX

LIMITED WARRANTY

K-BYTE™ will exchange this product within three months from date of purchase if defective in manufacture. Merchandise for exchange must be returned to K-BYTE™ along with proof of purchase. K-BYTE™ shall have no liability or responsibility to purchaser or any other person or entity with respect to any liability, loss or damage caused or alleged to be caused directly or indirectly by this product, including, but not limited to, any interruption of service, loss of business and anticipatory profits or consequential damages resulting from the use or operation of this product. This warranty gives you specific legal rights. You may also have other rights which vary from state to state.